UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/585,934	01/04/2007	Samuel Boutin	293409US2X PCT	9277

22850 7590 06/29/2009

OBLON, SPIVAK, MCCLELLAND MAIER & NEUSTADT, P.C.
1940 DUKE STREET
ALEXANDRIA, VA 22314

EXAMINER	
LEVIN, NAUM B	

ART UNIT	PAPER NUMBER
2825	

NOTIFICATION DATE	DELIVERY MODE
06/29/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

patentdocket@oblon.com
oblonpat@oblon.com
jgardner@oblon.com

Office Action Summary	Application No.	Applicant(s)	
	10/585,934	BOUTIN, SAMUEL	
	Examiner	Art Unit	
	NAUM B. LEVIN	2825	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 March 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *10-26* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *10-26* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *04 January 2007* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This office action is in response to application 10/585,934 and Response filed on 03/10/2009. Claims 1-9 have been canceled, claims 10-26 remain pending in the application.

Specification

2. This application does not contain an abstract of the disclosure as required by 37 CFR 1.72(b). An abstract on a separate sheet is required.

Applicant is reminded of the proper content of an abstract of the disclosure.

A patent abstract is a concise statement of the technical disclosure of the patent and should include that which is new in the art to which the invention pertains. If the patent is of a basic nature, the entire technical disclosure may be new in the art, and the abstract should be directed to the entire disclosure. If the patent is in the nature of an improvement in an old apparatus, process, product, or composition, the abstract should include the technical disclosure of the improvement. In certain patents, particularly those for compounds and compositions, wherein the process for making and/or the use thereof are not obvious, the abstract should set forth a process for making and/or use thereof. If the new technical disclosure involves modifications or alternatives, the abstract should mention by way of example the preferred modification or alternative.

The abstract should not refer to purported merits or speculative applications of the invention and should not compare the invention with the prior art.

Where applicable, the abstract should include the following:

(1) if a machine or apparatus, its organization and operation;

(2) if an article, its method of making;

(3) if a chemical compound, its identity and use;

(4) if a mixture, its ingredients;

(5) if a process, the steps.

Extensive mechanical and design details of apparatus should not be given.

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

The following title is suggested: method for design and verification of safety critical systems.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. Claim 10 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. A method that recites purely mental steps is descriptive material and is nor statutory if method **steps** not claimed as method (1) tied to another statutory class (such as a particular apparatus) or (2) transform underlying subject matter (such as an article or materials) to a different state or thing (See in re *Bilski,* 545 F.3d *943,* 88 USPQ2d 1385 (Fed. Cir. 2008)).

5. Claim 20 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. A computer program product comprising a

computer readable medium is descriptive material and is nor statutory (See MPEP 2106.01) if not claimed as a computer-readable **storage device** for storing computer-executable software/instructions/program because the computer program product comprising a computer readable medium is non-statutory subject matter per se (for example, a carrier wave).

Claim Objections

6. In claims 10, 20 and 23 Applicant must clarify what is "where possible" (what will be "where not possible"?).

7. Claim 21 is objected to because of the following informalities: replace "a method" with – a computer program product --.

8. Claims 24 and 26 are objected to because of the following informalities: replace "a method" with – a design tool --.

9. Claim 25 is objected to because of the following informalities: replace "a computer program product" with – a design tool --.

Appropriate correction is required.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

10. Claims 10-17, 19-26 are rejected under 35 U.S.C. 102(b) as being unpatentable by Torres-Pomales (Software Fault Tolerance: A Tutorial; Technical Report: NASA-2000-tm210616; Year of Publication: 2000; pages 1-55).

11. As to claims 10, 20 and 23 Torres-Pomales discloses:

Claim 10 A method of producing a system architecture comprising a plurality of electrical devices connected to each other, said system preferably comprising a fault tolerant system (page 2, paragraphs 2-3; page 34, last paragraph; page 35, paragraph 1; Fig.1), **the method including:**

a) identifying a set of undesirable events and ascribing to each of said undesirable events an indicator of their severity (A system safety design begins by performing modeling and analysis to identify and categorize potential hazards. This is followed by the use of analysis techniques to assign a level of severity of occurrence to the identified hazards – page 8, paragraph 4);

b) associating where possible each said undesirable event with one or more actuators of said system architecture (As best understood, upon detection of a computer or actuator failure/associating where possible each said undesirable event, control is passed to another computer based on a predetermined hand over sequence – page 37, last paragraph);

c) developing a functional specification of an initial architecture proposed for implementation of said system architecture, said functional specification of said initial architecture (The initial design of this flight control computer was a four by three configuration including hardware and software/ functional specification

dissimilarity in all the channels. Software diversity was to be achieved through the use of different programming languages targeting different lane processors. The final and current implementation uses only one programming language with the executable code being generated by three different compilers still targeting dissimilar lane processors. The lane processors are dissimilar because they are the single most complex hardware devices, and thus there is a perceived risk of design faults associated with their use – page 36, last paragraph) **including dataflow for and between components thereof, said components comprising for example sensors or actuators** (page 4, paragraph 2; page 35, Fig.19);

d) refining on said functional specification the fault tolerance requirements associated with the severity of each said undesirable event and issuing refined fault tolerance requirements of said functional specification(Figure 2 presents the Prototyping Process Model. This process model is appropriate for projects where the requirements are incompletely specified or when the developers are unsure whether a proposed design solution is adequate. The process begins with a requirements capture activity, followed by a quick design and build of a prototype or mock-up of the product. After analyzing the prototype, further refinements to the requirements are generated and the process begins again. This cycle activity not only helps develop the requirements, but it also helps the developers better understand the problem - page 3, paragraph 2);

e) producing replicates in said functional specification together with attached indicators of independence of said replicates, said indicators reflecting

said refined fault tolerance requirements (Multi-version fault tolerance is based on the use of two or more versions (or "variants") of a piece of software, executed either in sequence or in parallel. The versions are used as alternatives (with a separate means of error detection), in pairs (to implement detection by replication checks) or in larger groups (to enable masking through voting). The rationale for the use of multiple versions is the expectation that components built differently (i.e., different designers, different algorithms, different design tools, etc) should fail differently. Therefore, if one version fails on a particular input, at least one of the alternate versions should be able to provide an appropriate output - page 10, paragraphs 2-6; page 11; page 12, paragraph 1; page 17, paragraphs 3-4; page 18; pages 19-21);

f) defining a hardware structure for said system architecture, e.g. a series of electronic control units connected to each other by networks (Fig. 19);

g) mapping of said functional specification onto said hardware structure (The integration process is the phase of development when the source code is linked and transformed into the executable object code to be loaded on the target computer hardware - page 4, paragraph 4); **and**

h) verifying automatically that said indicators of independence are preserved during mapping (The supporting processes are three: verification, configuration management, and quality assurance. The purpose of the verification process is to search and report errors in the software requirements, design, source code, and integration. Verification is composed of three types of activities: reviews, analysis and testing – page 4, last paragraph, page 5);

Claim 20 A computer program product comprising a computer readable medium having thereon computer program code means, when said program is loaded, to make the computer execute procedure to design and verify system architecture (page 27, paragraphs 2-3), **said procedure comprising:**

a) identifying a set of undesirable events and ascribing to each of said undesirable events an indicator of their severity (A system safety design begins by performing modeling and analysis to identify and categorize potential hazards. This is followed by the use of analysis techniques to assign a level of severity of occurrence to the identified hazards – page 8, paragraph 4);

b) associating where possible each said undesirable event with one or more actuators of said system architecture (As best understood, upon detection of a computer or actuator failure/associating where possible each said undesirable event, control is passed to another computer based on a predetermined hand over sequence – page 37, last paragraph);

c) developing a functional specification of an initial architecture proposed for implementation of said system architecture, said functional specification of said initial architecture (The initial design of this flight control computer was a four by three configuration including hardware and software/ functional specification dissimilarity in all the channels. Software diversity was to be achieved through the use of different programming languages targeting different lane processors. The final and current implementation uses only one programming language with the executable code being generated by three different compilers still targeting dissimilar lane processors.

The lane processors are dissimilar because they are the single most complex hardware devices, and thus there is a perceived risk of design faults associated with their use – page 36, last paragraph) **including dataflow for and between components thereof, said components comprising for example sensors or actuators** (page 4, paragraph 2; page 35, Fig.19);

d) refining on said functional specification the fault tolerance requirements associated with the severity of each said undesirable event and issuing refined fault tolerance requirements of said functional specification(Figure 2 presents the Prototyping Process Model. This process model is appropriate for projects where the requirements are incompletely specified or when the developers are unsure whether a proposed design solution is adequate. The process begins with a requirements capture activity, followed by a quick design and build of a prototype or mock-up of the product. After analyzing the prototype, further refinements to the requirements are generated and the process begins again. This cycle activity not only helps develop the requirements, but it also helps the developers better understand the problem - page 3, paragraph 2);

e) producing replicates in said functional specification together with attached indicators of independence of said replicates, said indicators reflecting said refined fault tolerance requirements (Multi-version fault tolerance is based on the use of two or more versions (or "variants") of a piece of software, executed either in sequence or in parallel. The versions are used as alternatives (with a separate means of error detection), in pairs (to implement detection by replication checks) or in larger

groups (to enable masking through voting). The rationale for the use of multiple versions is the expectation that components built differently (i.e., different designers, different algorithms, different design tools, etc) should fail differently. Therefore, if one version fails on a particular input, at least one of the alternate versions should be able to provide an appropriate output - page 10, paragraphs 2-6; page 11; page 12, paragraph 1; page 17, paragraphs 3-4; page 18; pages 19-21);

f) defining a hardware structure for said system architecture, e.g. a series of electronic control units connected to each other by networks (Fig. 19);

g) mapping of said functional specification onto said hardware structure (The integration process is the phase of development when the source code is linked and transformed into the executable object code to be loaded on the target computer hardware - page 4, paragraph 4); **and**

h) verifying automatically that said indicators of independence are preserved during mapping (The supporting processes are three: verification, configuration management, and quality assurance. The purpose of the verification process is to search and report errors in the software requirements, design, source code, and integration. Verification is composed of three types of activities: reviews, analysis and testing – page 4, last paragraph; page 5);

Claim 23 A design tool configured for design and verification of a system architecture, the system architecture including a plurality of electrical components connected to each other, the components including electronic

control units, sensors, and actuators (page 27, paragraphs 2-3; page 28, paragraphs 2-9), **the design tool configured to:**

a) identify set of undesirable events and ascribing to each of said undesirable events an indicator of their severity (A system safety design begins by performing modeling and analysis to identify and categorize potential hazards. This is followed by the use of analysis techniques to assign a level of severity of occurrence to the identified hazards – page 8, paragraph 4);

b) associate where possible each said undesirable event with one or more actuators of said system architecture (As best understood, upon detection of a computer or actuator failure/associating where possible each said undesirable event, control is passed to another computer based on a predetermined hand over sequence – page 37, last paragraph);

c) develop a functional specification of an initial architecture proposed for implementation of said system architecture, said functional specification of said initial architecture (The initial design of this flight control computer was a four by three configuration including hardware and software/ functional specification dissimilarity in all the channels. Software diversity was to be achieved through the use of different programming languages targeting different lane processors. The final and current implementation uses only one programming language with the executable code being generated by three different compilers still targeting dissimilar lane processors. The lane processors are dissimilar because they are the single most complex hardware devices, and thus there is a perceived risk of design faults associated with their use – page 36,

last paragraph) **including dataflow for and between components thereof, said components comprising for example sensors or actuators** (page 4, paragraph 2; page 35, Fig.19);

d) refine on said functional specification the fault tolerance requirements associated with the severity of each said undesirable event and issuing refined fault tolerance requirements of said functional specification(Figure 2 presents the Prototyping Process Model. This process model is appropriate for projects where the requirements are incompletely specified or when the developers are unsure whether a proposed design solution is adequate. The process begins with a requirements capture activity, followed by a quick design and build of a prototype or mock-up of the product. After analyzing the prototype, further refinements to the requirements are generated and the process begins again. This cycle activity not only helps develop the requirements, but it also helps the developers better understand the problem - page 3, paragraph 2);

e) produce replicates in said functional specification together with attached indicators of independence of said replicates, said indicators reflecting said refined fault tolerance requirements (Multi-version fault tolerance is based on the use of two or more versions (or "variants") of a piece of software, executed either in sequence or in parallel. The versions are used as alternatives (with a separate means of error detection), in pairs (to implement detection by replication checks) or in larger groups (to enable masking through voting). The rationale for the use of multiple versions is the expectation that components built differently (i.e., different designers, different

algorithms, different design tools, etc) should fail differently. Therefore, if one version fails on a particular input, at least one of the alternate versions should be able to provide an appropriate output - page 10, paragraphs 2-6; page 11; page 12, paragraph 1; page 17, paragraphs 3-4; page 18; pages 19-21);

f) define a hardware structure for said system architecture, e.g. a series of electronic control units connected to each other by networks (Fig. 19);

g) map of said functional specification onto said hardware structure (The integration process is the phase of development when the source code is linked and transformed into the executable object code to be loaded on the target computer hardware - page 4, paragraph 4); **and**

h) verify automatically that said indicators of independence are preserved during mapping (The supporting processes are three: verification, configuration management, and quality assurance. The purpose of the verification process is to search and report errors in the software requirements, design, source code, and integration. Verification is composed of three types of activities: reviews, analysis and testing – page 4, last paragraph; page 5).

12. As to claims 11-17, 19, 21-22 and 24-26 Torres-Pomales recites:

Claims 11, 24 A method/design tool, wherein the system includes a fault tolerant system (page 2, paragraphs 2-3; page 34, last paragraph; page 35, paragraph 1; Fig.1);

Claim 12 A method includes in the developing step (c) defining a series of modes of operation (page 12, last paragraph; page 13; page 14, paragraph 1);

Claim 13 A method for defining a series of modes of operation include nominal and limp-home/degraded modes (page 17, last paragraph);

Claim 14 A method that includes specifying said series of modes in the form of one or more state charts (page 12, last paragraph; page 13; page 14, paragraph 1);

Claim 15 A method including mapping geometrically hardware components and/or wiring and then verifying automatically that said indicators of independence are preserved by said geometrical mapping (page 4, last paragraph; page 5);

Claim 16 A method including specifying severity in the form of probability of failure per unit of time (page 7, paragraphs 2-4; page 5);

Claim 17 A method according to any preceding claim including outputting a set of data for manufacturing said system architecture (page 26, paragraph 3; page 29, paragraph 1);

Claims 19, 21, 26 A method/program/tool, wherein the hardware structure is in a form of a series of electronic control units connected to each other by networks (Figs.19-21);

Claims 22, 25 A computer program product/tool, wherein the components include sensors or actuators (Figs.19, 21).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

13. Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable by Torres-Pomales in view of Ivarez-Troncoso et al. (U.S. Patent 7,076,350).

With respect to claim 18 Torres-Pomales teaches the features above but lacks a method, wherein a system architecture comprises a safety critical architecture for a vehicle.

14. As to claim 18 Ivarez-Troncoso recites:

Claim 18 A method, wherein the system architecture includes a safety critical architecture for a vehicle (electrical energy, power, and load management (EM) system controls power generation, storage, and consumption according to different operational states of a vehicle. An EM system typically has the capability of disengaging lower priority loads during specific conditions in order to maintain sufficient power to higher priority (e.g., safety-related) loads). Electrical system parameters that are monitored and controlled include break system of the vehicle - col.2, lines 34-67; col.3, lines 1-19).

It would have been obvious to a person of ordinary skills in the art at the time the invention was made to improve Torres-Pomales' invention by implementing the method,

wherein the system architecture includes a safety critical architecture for a vehicle by providing an energy management control system for managing events in an electrical system based on forecasted states, thereby avoiding degraded electrical system performance without excessive processing requirements (col.1, lines 49-64).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to NAUM B. LEVIN whose telephone number is (571)272-1898. The examiner can normally be reached on M-F (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jack Chiang can be reached on 571-272-7483. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Naum Levin/

Primary Examiner
Art Unit 2825